# TDARACNE

P. Zoppoli, S. Morganella, M. Ceccarelli

## Contents

## 1 Overview

This document describes classes and functions of TD-ARACNE (TimeDelay ARACNE) package.
One of main aims of Molecular Biology is the gain of knowledge about how molecular components interact each other and to understand gene function regulations. Using microarray technology, it is possible to extract measurements of thousands of genes into a single analysis step having a picture of the cell gene expression. Several methods have been developed to infer gene networks from steady-state data, much less literature is produced about time-course data, so the development of algorithms to infer gene networks from time-series measurements is a current challenge into bioinformatics research area. In order to detect dependencies between genes at different time delays, we propose an approach to infer gene regulatory networks from time-series measurements starting from a well known algorithm based on information theory.
We have shown (?) how the ARACNE (Algorithm for the Reconstruction of Accurate Cellular Networks) algorithm (?) can be used for gene regulatory network inference in the case of time-course expression profiles. The resulting method is called TimeDelay-ARACNE. It just tries to extract dependencies between two genes at different time delays, providing a measure of these dependencies in terms of mutual information. The basic idea of the proposed algorithm is to detect time-delayed dependencies between the expression profiles by assuming as underlying probabilistic model a stationary Markov Random Field. Less informative dependencies are filtered out using an auto calculated threshold, retaining most reliable connections. TimeDelay-ARACNE

can infer small local networks of time regulated gene-gene interactions detecting their versus and also discovering cyclic interactions also when only a medium-small number of measurements are available.
The idea on which TimeDelay-ARACNE is based comes from the consideration that the expression of a gene at a certain time could depend by the expression level of another gene at previous time point or at very few time points before. TimeDelay-ARACNE is a three-steps algorithm: first it detects, for all genes, the time point of the initial changes in the expression, secondly there is network construction and finally a network pruning step.
The goal of TimeDelay-ARACNE is to recover gene time dependencies from time-course data producing oriented graph. To do this we introduce time Mutual Information and Influence concepts. First tests on synthetic networks and on yeast cell cycle, SOS pathway data and IRMA give good results but many other tests should be made. Particular attention is to be made to the data normalization step because the lack of a rule. According to the little performance loss linked to the increasing gene numbers shown in (**?**), next developmental step will be the extension from little-medium networks to medium networks.
TD-ARACNE algorithm given time-course gene expression values allows to obtain an oriented graph representing a gene regulatory network. The goal of this package is to create a tool useful to gene regulation's researchers in order to obtain a first unvalidated look at the regulatory network. The package makes use of the GenKern (**?**) package to compute the kernel, the Biobase package (**?**) and the Rgraphviz package (**?**). The functions in the package will work on numeric data organized in a matrix. The results of these procedures will change slightly depending on normalization choice and the number of the classes in the discretization step. The example data included in the package are used in (**?**)

## 2 Data Description

Input data is an ExpressionSet object. An example can be a dataset downloaded from GEO or ArrayExpress converted in an ExpressionSet object . In the TDARACNE package you can found 3 example datasets. The first one is the Yeast dataset, a time course profile made by a set of 11 genes, part of the G1 step of yeast cell cycle, selected from the widely used yeast, *Saccharomyces cerevisiae*, previously published by **?** for which 16 time points are available. The data can be loaded as follows:

```
> data(dataYeast)
> data(threshYeast)
```

The second dataset is made by the time course profiles for a set of 8 genes, part of the SOS pathway of *E. coli* **?** from which the first 14 points (excluding the first point of the data which is zero) are used.

```
> data(dataSOSmean)
> data(threshSOSmean)
```

The third one is a 5 genes sets of time course profiles provided by real-time PCR from an in vivo yeast synthetic network **?**. The Switch ON data set, is the result of the time measurements, every 20 minutes for 5 hours, of the mRNA concentration after shifting cells from glucose to galactose, for a total of 5 profiles of 16 points.

```
> data(dataIRMAon)
> data(threshIRMAon)
```

# 3 main function

The main function is: TDARACNE(z, N, name, delta , likehood, norm, logarithm, thresh, ksd, tolerance) Here follow a brief description of the arguments: arguments **eSet**: eSet is the ExpressionSet object

**N**: N is respectively the number of bins in percentile normalization or in rank normalization

**delta**: delta is the maximum time delay allowed to infer connections

**likehood**: likehood is the fold change used as threshold to state the initial change expression (IcE)

**norm**: normalization;
if you want column percentile normalization (row normalization) put norm == 1;
if you want Rank normalization put norm == 2;

**logarithm**: if z is log put logarithm == 0;

**thresh**: the Influence threshold.
if you have a threshold and a SD( standard deviation) put them here in this format: c(thresh,SD);

if you don't have threshold put 0 in thresh;
**ksd**: ksd is the standard deviation multiplier;
**tolerance**: tolerance is the DPI tolerance;
0 means no tolerance;
1 means no DPI;
0.15 is the default ARACNE tolerance as it is for TDARACNE;
**plot**: plot must be TRUE to obtain automatically the graph
**dot**: dot must be TRUE to obtain a .dot file
**name**: the name of the .dot file resulting in the end
**adj**: adj must be TRUE to obtain an adjacent matrix

TDARACNE() automatically load the libraries that needs but it requires them installed when TDARACNE() starts. An example using the embedded dataset is:

```
> TDARACNE(dataIRMAon,11,"netIRMAon",delta=3,likehood=1.2,norm=2,logarithm=1,thr
```

To obtain all the results published in the paper **?** you can use (attention is time consuming):

```
>  TDARACNEdataPublished()
```

# 4 Output File Format

At the end of the computation TDARACNE() returns a graphNEL object, an adjacent matrix, a graph or a .dot file according to the parameter selected. By default it returns a graphNEL object.